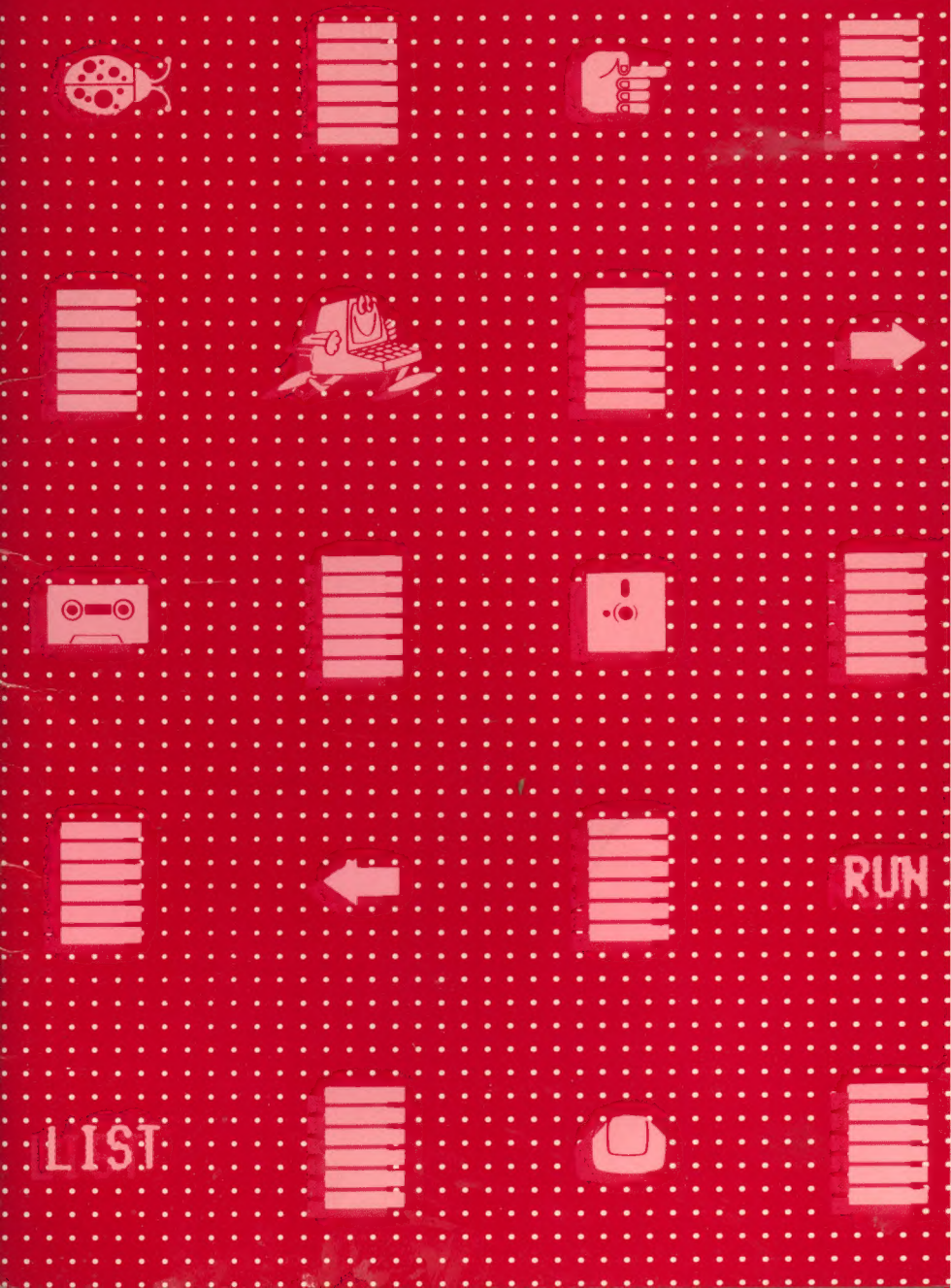
A·C·A·D·E·M·Y

ON

COMPUTERS

HANDS-ON ATARI 400/800 A BEGINNER'S MANUAL



Academy on Computers

HANDS-ON ATARI 400/800 A BEGINNER'S MANUAL

WNET/THIRTEEN P.O. Box 813 New York, NY 10101

This manual is part of the Academy on Computers, a self-directed learning activity. The Academy is based on the *Bits and Bytes* programs produced by the TVOntario network and presented in the United States by WNET/THIRTEEN.

Project Team in Canada:

Project Officer: Judy Winestone
Editor: Mei-Lin Cheung
Designers: Danny Leung/Roswita Busskamp
Photographer: Garth Scheuer

This hands-on manual follows the model developed by Trudy Van Buskirk for other microcomputers used in The Computer Academy. The adaptation for the ATARI 400/800 was made with the assistance of Lawrence Breakwell, founder and president of the Toronto Atari Programmers' Society.

Project Team in United States:

Director: Carol L. Angert
Supervisor of Academy Services: Peggy Yalman
Supervisor of Station Relations: Lois Dino
Project Assistant: Denise O'Connor

WNET Administrative Staff
Director of Educational Programs: Shirley B. Gillette
Senior Manager of Educational Programs: Debbi Bilowit
Publications Supervisor: Melinda Klaber
Manager of Systems Analysis: Ian Coles
Vice President and Director, Education Division: Stephen L. Salyer

Academy on Computers is presented in the United States by the Education Division of WNET/THIRTEEN.

Academy on Computers was conceived, developed, and produced by TV-Ontario, Canada's leading educational television service.

TVOntario

ORDERING INFORMATION

For more information, please write to:
Academy on Computers
WNET/THIRTEEN
P.O. Box 813
New York, NY 10101

Printed in the United States.

Logo Design: Juanita Gordon

TABLE OF CONTENTS

NOTE:
Many people write computer programs that they are happy to give away free. There are thousands of them freely circulating in user groups, by mail, or in newsletters. Some of these programs are very good indeed, some are small and simple, and some don't work.

Commercial programs, on the other hand, are designed for sale. They are professionally written, often large and powerful, and they come with good documentation. When you pay for a product, you expect to get more, and in the field of computers you usually do.

It's often possible for an individual to make copies of a commercial program to give or sell to others. This is called piracy; it's highly unethical. The user receiving the pirate copy will lose on documentation and support; the pirate is liable to be sued; and the commercial program house may give up producing and selling programs that are pirated. We all lose. Remember that in all cases the onus is on the person making a copy of a program to ensure that copyright interests are not being violated.

BEFORE YOU BEGIN

INTRODUCTION

This book is intended to be used in a hands-on experience with an ATARI 400 or 800 Home Computer. It introduces you to the machine and takes you step by step through a variety of activities designed to increase your understanding and command of the ATARI. It does not presume any prior knowledge of computers, and avoids jargon wherever possible.

As you will discover in going through the book, there is no magic in learning how to operate a microcomputer. There are, however, a couple of strategies that will make your learning experience more successful: do work through all of the suggested activities, and be patient with yourself. For many of you, this is probably a very new kind of experience and it may take you a while to catch on. When you do, though, we think you'll find that computers can be a lot of fun!

We also hope that this book and your hands-on experience will aid you in evaluating the role of microcomputers in society. In a world where most of us encounter a computer in some form almost every day, it's important that we all have at least an elementary grasp of what computers are and what they can and cannot do.

OBJECTIVES OF THE HANDS-ON MATERIALS

When you have finished working through the hands-on materials, you should be able to:

- operate the ATARI 400 or 800 Home Computer and cassette recorder;
- load and run programs from cassette;
- save and copy software;
- use the computer as a calculator;
- write short programs using BASIC commands; and
- modify BASIC programs.

This manual is only a beginning. Once you are comfortable with the ATARI, you can quickly and easily master the operation of other microcomputers simply by reading the operation instructions that accompany the hardware. Microcomputers do not differ markedly from one another, and you are learning about features that are common to them all.

WHAT YOU NEED TO GET STARTED

Hardware:

You need an ATARI 400 or 800 Home Computer System with 16K (or larger) of memory. You also need a display of some sort — either a standard TV or video monitor (black-and-white or color) or a video monitor (used only with the ATARI 800), and an ATARI 410 Cassette Recorder.

Software:

In addition to the cassettes containing software which accompany this manual, you will need the ATARI BASIC cartridge. You will also need one ten-minute blank cassette tape.

Time:
You will take approximately 60 minutes to complete each of the six units in this manual. Although you are free to work through the activities at your own pace, you will learn more and work more effectively if you spread the work evenly over the duration of the Academy. Remember: you are in charge of the learning — not the computer. Don't hesitate to stop if you become tired or frustrated.

THE BOOK ITSELF
Units I and II introduce you to the ATARI keyboard. You will learn what each of the keys does and how to correct typing errors. You will also learn how to load, run, copy, and save ready-made programs.

Unit III shows you how to use the computer as a calculator and how to list the program on the sample cassette.

Units IV and V teach you elementary programming in BASIC — not enough for you to write a very useful program, but enough for you to understand what programming is and to go on to more advanced texts and courses, if you wish.

Unit VI explains how to modify existing programs to suit your own needs.

The section entitled "Troubleshooting" identifies some of the more common programming errors and explains how to clear the screen, stop a program, start again, etc.

The glossary contains explanations of many of the technical terms used in this manual. In the text, we have tried to keep these to a minimum, and to explain them as we go along.

UNIT I
HOW DO YOU START?

Let's begin this first hands-on session in a very methodical way.

The ATARI consists of at least three separate components: the keyboard, display unit, and a cassette tape recorder.

Check your ATARI 400 or 800 owner's manual for proper set-up and installation instructions. Make sure that all cords and power supplies are plugged in according to the instructions.

Open the cartridge door and place the ATARI BASIC cartridge in the 400's single slot (or the left cartridge slot of the 800). Close the door. Plug the 410 Cassette Recorder into the side of the computer.

Turn on the TV and tune it to either channel 2 or 3 for best reception. Then move the button by the on/off power switch to indicate the channel you are using.

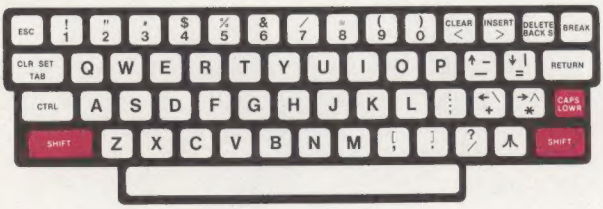
You are now ready to turn on the computer. The on/off switch is on the right side of the computer console.

The computer will proceed to load the BASIC program.

When this is done, a blue screen with a black border will be displayed and the following message will appear:

On the line following the "READY" message will be a white square known as the "cursor." It indicates that the computer is waiting for the user to give it instructions. (You may wish to adjust the brightness and contrast of the screen to make the message easier to read.)

THE KEYBOARD



The keyboard is like a standard typewriter, with keys of several types. Pressing any key causes its upper-case character to display on the screen, and pressing the CAPS LOWR key causes the lower-case characters to appear. When in lower-case mode, pressing the SHIFT key will cause the upper-case character to appear for that key.

Some of the keys also have special functions.

ALPHABET KEYS

All alphabetical symbols are typed into the computer using the alphabet keys. To capitalize letters, hold down the SHIFT key and press the letter key, just as you would on a standard typewriter keyboard.

SPECIAL KEYS

SPACE BAR — The space bar is used to type "blanks" — spaces between words and numbers, for example. To a computer, a blank is as much a "character" as the letters A, B, and C. The space bar can be used to move the cursor to the right, but — and this is where it differs from the space bar on a typewriter — it will replace any character it encounters with a blank.

ESC ESCAPE — The ESC key is used in conjunction with the CTRL key and SHIFT key to enable special functions, and the printing of special graphics characters.

CLR-SET TAB CLEAR-SET-TAB — The CLR-SET TAB key allows you to set and clear tab positions. Used alone, this key advances the cursor to the

next preset tab position. (Defaults — that is, normal tab settings — are 15, 23, 31, and 39.) Pressing the SHIFT and CLR·SET TAB keys simultaneously will set a tab at the current cursor position.

To clear a tab, move the cursor to the tab position you wish removed. Press the CTRL and CLR·SET TAB keys simultaneously and the tab will be removed.

CTRL
CONTROL

The CTRL key is always used in conjunction with other keys to perform certain functions. On keys having three functions, their use with the control key will produce the graphics characters. (For now just notice where it is located.)

SHIFT

There are two SHIFT keys, one on each side of the keyboard. They both serve the same purpose. The SHIFT key enables you to type upper-case characters (i.e., capital letters) and to type the symbols that appear above the numbers in the top row of the keyboard.

NOTE: The ATARI accepts commands only in UPPER CASE.

CLEAR

Pressing the CLEAR key simultaneously with either the SHIFT or CTRL key will completely clear the screen and put the cursor in the upper left corner of the screen.

INSERT

If the SHIFT and INSERT keys are pressed simultaneously, the computer will insert one blank line above the present line and return the cursor to the left margin.

NOTE: No line is deleted in this process. Pressing the INSERT and CTRL keys together will insert a single blank space in a line.

DELETE BACK S
DELETE
BACK
SPACE

The DELETE BACK S key removes the characters from the screen. Pressing the DELETE BACK S key replaces the character to the left of the cursor with a space and moves the cursor back one space. Using this key with SHIFT deletes the entire line. Using it with CTRL deletes the single character behind the cursor.

BREAK

The BREAK key will stop execution of the program that is running.

RETURN

The RETURN key enters what you have typed into the computer. A numbered line will be interpreted (or put into terms the computer can execute) by the BASIC cartridge program and added to your program.

An unnumbered line is interpreted and executed immediately.

CURSOR
CONTROLS

When the CTRL key is pressed along with one of the cursor control (arrow) keys, the cursor will move one space in that direction.

NOTE: The cursor will move without erasing any of the characters it moves over.

CAPITALS LOWER

The CAPS LOWR key "locks" the keyboard enabling you to type lower-case characters. The CAPS LOWR key does not affect the number keys or symbol keys. To "unlock" the keyboard, simply press the SHIFT and CAPS LOWR keys together. You can now type upper-case characters.

ATARI LOGO

The (Reverse or Inverse Video, or "ATARI LOGO") key if pressed will cause the text to be reversed on the screen (dark text on light background). Pressed again, the text will be restored.

MATH OR OPERATION KEYS

The math (or operation) keys are used to instruct the computer to add, subtract, multiply, divide, and raise a number to a power (exponentiate).

The symbols for addition, subtraction and equals are the ones with which we are familiar, but the multiplication and division symbols are new.

The Atari Personal Computer System uses five arithmetic operators:

- + addition
- − subtraction
- * multiplication
- / division
- ^ exponentiation

Before going on, have some fun with the keyboard. Fill the screen with characters. Try out all the keys. Press them one by one; hold one key down longer and see what happens. Experiment with the SHIFT, CAPS LOWR, CTRL, CLEAR, INSERT, DELETE BACK S, and  keys.

Don't be afraid to play with the keyboard. That's what children do and they learn very quickly. Remember: you can't break a computer unless you physically drop or throw it — something you may want to do before you have finished working through this manual.

When you have completely filled the screen with characters, count the number of characters and spaces in a single line and then the number of rows on the screen.

There should be 38 characters (including spaces) across and 24 rows down. These are the default dimensions of the screen. The margins can be expanded or contracted using special functions. The maximum dimensions are 40 x 24.

If you wish, you can reduce the width of the screen.

First, however, you should clear the screen. To do this, either

a) Type GRAPHICS 0 and press RETURN .

or

b) You could "RE-BOOT" the system by turning the computer off and on again.

or

c) Simply hold down the SHIFT key and press the CLEAR key. The screen will be cleared and the cursor placed at the left hand margin of the screen, at the top. The "READY" message will not be printed but the cursor indicates the computer is waiting for your next commands.

Pressing CTRL and CLEAR will accomplish the same as the SHIFT and CLEAR routine.

Procedures a), b), and c) will produce this screen:

To change the width of the screen is very simple. All that must be done is to "POKE" the left and right margins into the computer's memory.

POKE means to place or put into the computer's memory a number. The computer will look at those memory locations and set the margins accordingly.

To change the left margin type POKE 82,x , where x is the column number you wish to have as the new margin, and press RETURN . To change the right margin it's POKE 83,x .

Go ahead and experiment. If you get into trouble just type GRAPHICS 0 and press RETURN to return to the default margin.

Let's move the left margin from its default position of column 2 to column 10. Type POKE 82,10 and press RETURN .

The cursor will now be positioned at column 10.

Let's move the right margin from its defaults of column 39 to column 30.

Type POKE 83,30 and press RETURN . (The screen will not visibly change.)

To check that you have successfully changed the margin of the screen, type THIS IS AN ATARI HOME COMPUTER SYSTEM.

(Don't worry if you make mistakes; you will soon be learning how to correct typing errors.)

To change back to the default parameters of the margin, type:

1. POKE 82,2 and press RETURN .
2. POKE 83,39 and press RETURN .

Notice that when the first line was filled, the computer automatically began to print on the second line. You do not have to tell the computer, as you would a typewriter, when to go to the next line.

HOW TO CORRECT TYPING ERRORS

Typing errors are much easier to correct on an ATARI than on a regular typewriter. Try this:

1. Type THUS IS A ATARI PERSONNAL COMPUTER

Your task is to change *THUS* to *THIS*, *A* to *AN*, and *PERSONNAL* to *PERSONAL*. Let's begin with PERSONNAL.

2. Use the cursor control keys CTRL and ↑ ↓ ← → to position the cursor under the second N.
3. Press the CTRL and DELETE BACK S keys. The word Personal should now be spelled correctly.

To change *A* to *AN*:

4. Use the cursor control keys to position the cursor in the space to the right of A.
5. Press the CTRL and INSERT keys. (Note that the cursor becomes a white box.)
6. Type N. Notice that all of the other characters in the line move over one space to make room for the new letter.

To change *THUS* to *THIS*:

7. Use the cursor control keys to position the cursor under the U.
8. Type I. The word THIS should now be correct.

Let's insert the word *HOME* between ATARI and PERSONAL.

1. Use the ↑ ↓ ← → to position the cursor over the P in PERSONAL.
2. Press CTRL and INSERT 5 times. This will make room for you to type in HOME.
3. Now type HOME.

That's all there is to it.

To remove *HOME*:

1. Use the ↑ ↓ ← → to move the cursor to the space just after the I in ATARI.
2. Press CTRL and DELETE BACK S 5 times.

The line is now restored.

Experiment with the keyboard and the functions you have learned.

A SAMPLE PROGRAM

Now that you know what the keyboard is all about and how to correct typing errors, try typing in this short program.

1. First, clear the screen by pressing CTRL and CLEAR. Then type NEW and press RETURN.
2. Type 10 PRINT "HELLO. I AM YOUR NEW ATARI."
3. Check your typing. Make sure that you have copied the line exactly, including the quotation marks.
4. When you are satisfied that the line is correct, press RETURN.
5. Type 20 PRINT "WHAT IS YOUR NAME?"
6. Check your typing and then press RETURN.
7. Type 30 DIM A$(30)

Check your typing; then press RETURN. This line reserves a place in memory, known as A$, 30 characters long.

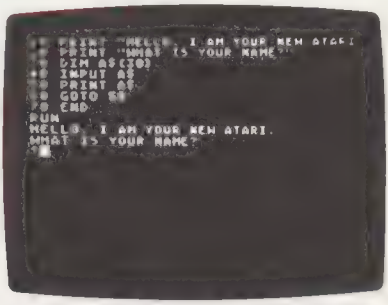
8. Type 40 INPUT A$

Again, check your typing; then press RETURN .

9. Type 50 PRINT A$

Again, check your typing, then press RETURN .

10. Type 60 GOTO 50 and press RETURN .
11. Type 70 END and press RETURN .
12. Type RUN and press RETURN . You will see this on the screen:
13. Type your name and then press RETURN .



14. To stop the program, turn off the machine. (In the next unit you will learn how to stop a program without having to turn off the machine.)

If an error message — "ERROR — 17 at Line x" or "ERROR — 9 at Line x" for example — appears on the screen, you can be fairly certain that you have made a typing error. Type NEW and then press RETURN . Then retype the program.

SUMMARY

The ATARI keyboard consists of both alphabetic and number keys, as well as "special" keys such as RETURN , CTRL , CLEAR , INSERT , DELETE BACK S , CAPS LOWR , and . There are also four cursor control keys that allow you to position the cursor anywhere on the screen.

The maximum dimensions of the screen are 40 characters by 24 rows. You can, however, reduce the width of the screen. The default dimensions the computer sets up are 38 by 24.

There are FOUR ways to clear the screen: (1) by turning the computer off and on again (this empties the computer's memory); (2) by typing GRAPHICS 0 and pressing RETURN ; (3) by pressing SHIFT and CLEAR or (4) by pressing CTRL and CLEAR .

UNIT II
HOW TO USE READY-MADE PROGRAMS

HOW TO LOAD AND RUN A PROGRAM FROM A CASSETTE

Running a computer cassette recorder is much like running an audio tape recorder. The tapes themselves are made of the same magnetic material; also, the buttons on both devices are identical. The difference lies in the fact that an audio tape recorder receives its instructions to play the tape through depression of the appropriate sequence of buttons: rewind then play. The computer tape recorder, on the other hand, receives its instructions first from the computer, then from the buttons. Turn the computer on as you learned in UNIT I. You will see the same "READY" message and the cursor flashing every time you turn on the ATARI.

LOADING READY-MADE PROGRAMS

Now you are ready to load a program from the cassettes that accompany this manual.

1. Take a cassette that accompanies this manual out of its case. Open the cover of the cassette recorder and carefully slide the cassette into the recorder. Then close the cover.
2. To load the program from the cassette, type CLOAD and press RETURN . This command tells the computer to look for a program on the cassette. After pressing RETURN there will be a beep from the computer. This indicates that the PLAY button on the cassette recorder should be pressed. However, before pressing the PLAY button the tape should be rewound to the desired position (i.e., the point on the tape at which the program begins). After this has been done, the PLAY button should be depressed, followed by RETURN .

   The computer will load the program from the tape into its memory. There will be a series of beeps on the video display unit while the program is loading.

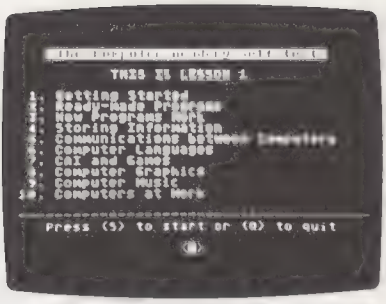
If the program loads, the READY message will appear on the video display unit screen.

If the program does not load, rewind the tape to the beginning and try the same procedure once again. The positioning of the tape is critical. If you continue to have difficulty in loading the program, rewind the tape, *manually* move the tape forward slightly and try again.

If you continue to have difficulty loading the program, ask a technician for help. Beginners often blame themselves when things don't work out as they should, but sometimes it's the cassette that is to blame or the recorder which is not working correctly.

HOW TO RUN THE PROGRAM NOW IN MEMORY

1. To run the SELFTEST program now in memory, type RUN and press RETURN .



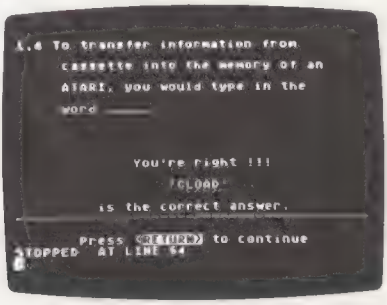
The SELFTEST program is a computer version of the self-tests that appear in the *Academy on Computers Resource Book.* We think you'll find the computer version much more interesting than the printed self-tests, and we encourage you to use the SELFTEST program on a regular basis.

2. Before going on, work through Self-test 1, "Getting Started," based on Chapter 1 in *Bits and Bytes.* When you've finished the test, read the next section of this manual which explains how to stop a program.

HOW TO STOP A PROGRAM — OR, "I DON'T WANT TO RUN THIS ANYMORE"

For any number of reasons, you may want to stop a program in the middle of its run. Perhaps you've forgotten the instructions and want to go back to the beginning; perhaps you're tired and want to quit; or perhaps you're just ready to go onto something new.

1. To stop a program in the middle of its run, press the BREAK key. (The BREAK key is located in the upper right corner of the keyboard.)



Each instruction or "line" in a program has a number that identifies it. In the example above, we stopped the program at line 64. The number you see on the screen will probably be different, depending on when you pressed the BREAK key.

At this point, you have three choices: (1) to continue from where you left off; (2) to start again from the beginning; or (3) to clear the program from the computer's memory.

1. To continue from where you left off, type CONT and press RETURN .
2. To start again from the beginning, type RUN and press RETURN .
3. To clear the program from the computer's memory, type NEW and press RETURN . Then press SHIFT and CLEAR to clear the screen.

With many commercial programs, it is not possible to "break in." Programmers can write programs that disable the BREAK key, i.e., make it nonfunctioning. This tactic is especially common in writing educational programs for young children.

To find other programs on a cassette locate the number on the counter of the recorder at which the program begins. Then follow the instructions on how to load programs from a cassette described earlier in this Unit.

HOW TO SAVE PROGRAMS

Many computer operators like to make second or "duplicate" copies of software programs, in the event that the first is destroyed or damaged. You should know that computer programs, like all other forms of "intellectual property" (e.g., books and articles, recorded music, videotapes, etc.) are protected by copyright. You are not permitted to copy programs without first obtaining permission from the copyright holder. In some cases, you may also be required to pay a fee.

The program that you are going to copy and save in the following exercise is the property of the Ontario Educational Communications Authority, which grants you permission to copy the program.

NOTE: To complete this exercise, you will need a blank cassette in addition to the cassettes that accompany this manual.

To save a program from the computer to cassette you have to type CSAVE. This command tells the computer that you are ready to save a program. When you are ready, type CSAVE.

Note that the computer will "write over" an existing taped program. In order to avoid this, the tape should be rewound to a blank (unused) section.

Rewind the cassette to the point where you want to save the program. After rewinding, remember to record the counter number.

After rewinding, press RETURN . You will hear two beeps this time. The computer is telling you that you must press two buttons on the recorder. Press RECORD and PLAY simultaneously. Then press RETURN on the computer.

The computer will start to save on to the cassette. After the computer has completed saving the program, and the "READY" message is displayed, press STOP on the cassette recorder and press RETURN on the computer.

1. Then rewind the cassette.
2. Replace the cassette with a new blank cassette.
3. Type CSAVE and press RETURN . When the computer has finished saving the SELFTEST, the cursor will appear on the screen.

SUMMARY

To run a program, type RUN and press RETURN . To stop a program in the middle of its run, press the BREAK key.

To load a program from cassette, type CLOAD.

To clear a program from the computer's memory, type NEW and press RETURN .

To save a program from one cassette to another, type CSAVE.

UNIT III
PROGRAMMING — WHO ME?

Fundamentally, programming is problem solving. You can program a computer to calculate your income tax, turn off your oven at a specified time, or determine the best route between Toronto and New York.

In solving any problem, you need to think about the steps that must be taken in order to reach a solution *and* about the order of those steps. Computers cannot help you with this. But once the steps have been identified, computers can help you with the solutions. However, computers need very precise and specific instructions, written in a code or language they can understand. This is what is meant by programming.

Of course, you don't have to become a programmer in order to operate a computer. Higher-level languages and new developments in hardware and software have democratized computer access. The majority of users now spend most of their time running ready-made programs. For most of us, it is enough that we can tell the computer how to do the things we want it to do.

USING THE COMPUTER AS A CALCULATOR

Any computer can be used as a calculator. Of course, if this were all it could do, it would be a very expensive calculator. But you have already seen evidence of some of the computer's other capabilities.

When you use a computer as a calculator, you are engaged in programming; that is, you give the computer a set of instructions which it carries out sequentially in order to solve a problem.

NOTE: In the exercise and activities that follow, remember that the (Ø) is different from the letter (O), and that the number (1) is different from the letter (l).

1. Type ? 2+7 and press RETURN .
2. Type ? 4+7-2 and press RETURN .
3. Type ? 25/5 and press RETURN .
4. Type ? 1Ø*3 and press RETURN .

A word about mathematical notation: While the + and − are quite straightforward, the * and / represent multiplication and division, respectively. Note that you do not type = at the end of a problem.

The question mark is another form of the BASIC command PRINT.

1. Type ? (2 + 3)*4–2 and press RETURN .
2. Type PRINT (2 + 3)*4–2 and press RETURN .

The answer to both problems is 18, since ? and PRINT mean the same thing.

The computer performs arithmetic functions in the following order: (1) brackets, (2) multiplication or division (whichever comes first), and (3) addition or subtraction (whichever comes first).

1. Press SHIFT and CLEAR together to clear the screen. (Alternatively, hold down the CTRL key and press CLEAR to clear the screen.)
2. Type ? (2 + 3)*4–2 and press RETURN .
3. Type ? 2 + 3*(4–2) and press RETURN .
4. Type ? 2 + (3*4)–2 and press RETURN .

Is this what you see on the screen?

Using a programming command (? or PRINT) to do arithmetic is called "immediate execution" mode. This means that the computer executes the command right away. It doesn't have to go through any other instructions before it prints the answer, the way it does in a program.

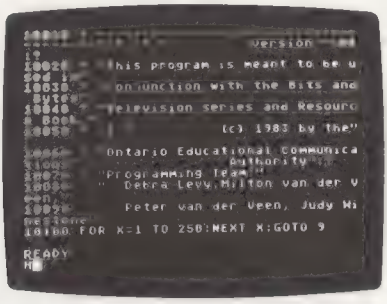
1. Type:
   ? 2 + 3*4/6*(12 + 6)–(13*2) + 6/10 + 5–4/3*(10*10)*15 + 25
   and press RETURN .

Is your answer -1957.399995? If it isn't, check your typing.

Before going on, try some arithmetic problems of your own. Remember: type ? first, and don't use = . Simply press RETURN when you want the answer.

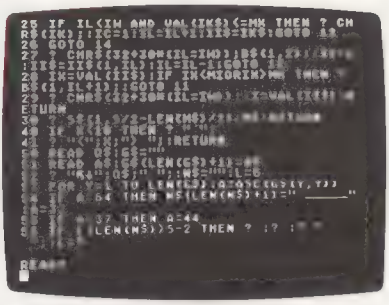
WHAT DOES A "BASIC" PROGRAM LOOK LIKE?

1. Put a cassette that accompanies this manual into the recorder.
2. Load the program called SELFTEST: type CLOAD and press RETURN . When you her a "beep", press PLAY on the cassette recorder and press RETURN on the computer.
3. Then, type LIST and press RETURN . Numbers, words, and symbols will begin to scroll on the screen. This is a "listing" of the program. When the program has reached the end, the "READY" message and cursor will appear on the screen.



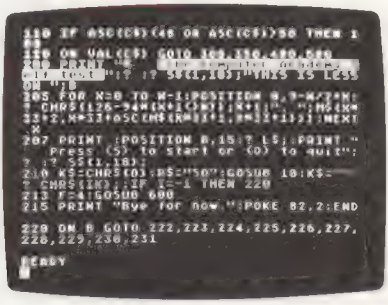
You can also stop the listing as it is scrolling.

4. Type LIST and press RETURN .
5. Hold down the BREAK key.



You can also choose to list any section of the program.

6. Type LIST 110, 220 and press RETURN .



Look carefully at the lines on the screen. Notice that the lines are numbered sequentially (e.g., 110, 120, 130, etc.). You'll see words like GOTO, REM, FOR, NEXT, etc. These are BASIC commands. In the next unit, we'll look more closely at some of these.

SUMMARY

A program is a set of instructions that tells the computer how to solve a specific problem. Programs are written in computer languages — artificial languages that have their own vocabularies and syntax. BASIC is the language used by most microcomputers. It stands for Beginners All-purpose Symbolic Instruction Code.

To see what a computer program looks like, type LIST and press RETURN . You can stop a listing at any time by pressing BREAK . You can also ask the computer to list specific sections or lines.

UNIT IV
PROGRAMMING IN BASIC (1)

CHARACTERISTICS OF BASIC PROGRAMS

All BASIC programs share the following characteristics:

- They are made up of lines called "statements."
- Each statement begins with a number.
- There is always an END statement.

List the program titled "SELFTEST."

Notice that the lines in the program are not numbered consecutively, (100, 101, 102, 103, etc.). The reason is that it's much easier to modify a program later if you leave room to insert new lines. It is traditional to space the lines in a program 10 numbers apart, (100, 110, 120, 130, etc.).

GIVING THE ATARI INSTRUCTIONS

A BASIC program is a set of instructions that tells the computer to do something. These instructions fall into two distinct categories: program commands and system commands.

System commands — for example, LIST, RUN, NEW, — tell the computer to do something *to* or *with* a program.

Program commands — for example, PRINT, GOTO, FOR . . . NEXT — appear only *within* the context of a program.

PRINT

The PRINT statement is simple enough to understand, but powerful enough to enable the computer to execute complex arithmetic operations.

1. Type NEW and press RETURN .
2. Type 10 PRINT 4+4 and press RETURN .
3. Type 20 END and press RETURN .

In order to execute this short program, you need to use a system command.

4. Type RUN and press RETURN . You should see this:

The correct answer is 8. Notice that the "READY" message is displayed. This tells you that the computer has finished the program and is waiting for you to enter new information.

NOTE: We hope that by now using the RETURN key is second nature to you. From this point on, we will no longer remind you to press the RETURN key after typing a command or statement.

In the next example, we have deliberately misspelled the word PRINT in line 10 in order to demonstrate how to correct errors in a program. Type:

```
NEW
10 PRUNT 4+4
20 END
RUN
```

An error message will appear on the screen after you enter line number 10.

The "ERROR —" is the computer's way of telling you to retype the line. Type:

```
NEW
10 PRINT 4+4
```

Now, complete and run the program.

Try this. Type:

```
05 GRAPHICS 0
10 PRINT "4+3="
20 PRINT 4+3
30 END
```

In this program, lines 10 and 20 illustrate two different uses of the command PRINT. In line 10, it tells the computer to print on the screen whatever information appears within quotation marks. In line 20, it tells the computer to calculate the sum of 4+3. Notice that the main difference between the two lines is the quotation marks that appear in line 10. Line 05 is also important: GRAPHICS 0 tells the computer to clear the screen first and set up the screen in 38 by 24 characters mode. Now RUN the program.

Type NEW and then press RETURN . Then type:

```
05 GRAPHICS 0
10 PRINT "TWO TIMES SEVEN IS"
20 PRINT 2*7
30 END
RUN
```

Here are a few more programs for you to try. Type NEW and press RETURN . Then type:

```
05 GRAPHICS 0
10 PRINT "***WOW***"
20 END
RUN
```

Type NEW and press RETURN . Then type:

```
05 GRAPHICS 0
10 PRINT "******"
20 PRINT "*     *"
30 PRINT "*"
40 PRINT "*"
50 PRINT "*     *"
60 PRINT "******"
70 END
RUN
```

To insert a line in the middle of this program, type:

```
25 PRINT "*"
LIST
```

Notice that line 25 appears in the correct place in the numeric sequence, that is to say, between lines 20 and 30.

If, instead of inserting a line, you wanted to delete a line, you would type:

40 (press RETURN)
LIST

To delete any line from a program, simply type the line number without any information beside it.

No matter how you enter lines in a program, or add or delete lines, the computer always arranges them in the correct numeric sequence.

VARIABLES

Memory is a collection of storage locations (similar to mailboxes), each with a unique address. In BASIC, a programmer can store information in a memory location without knowing or caring where that is. The information stored in a specific memory location can then be referred to by a symbolic name.

Information referenced in this way is known as a "variable" and the symbolic name is known as a "variable name." Variables are used to represent information (e.g., a word, a number, a name, an address) that the computer gets from the user in response to a question (e.g., What is your favorite number?).

There are two kinds of variables: numeric and string. A numeric variable represents any number. Numeric variables can be written as letters, letter-number pairs, or letter-letter pairs, (e.g., A, AB, A3). A string variable represents a group of numbers or letters or any combination of the two. String variables are written in the same way as numeric variables, but with a $ at the end, (e.g., A$, AB$, A3$). Both numeric and string variables must begin with an alphabetic character.

INPUT

An input statement allows the computer to ask for information from the user. A string variable appears in the input statement when a word is wanted; a numeric variable, when a number is wanted.

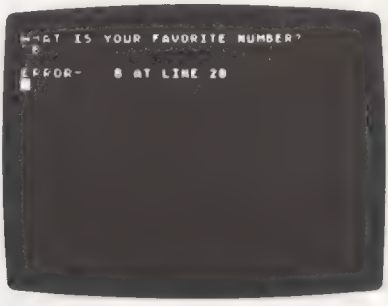
In the following example, A is a numeric variable in the input statement because the question in line 10 asks the user to supply a number. Type:

```
NEW
05 GRAPHICS 0
10 PRINT "WHAT IS YOUR FAVORITE NUMBER?"
20 INPUT A
30 PRINT "YOUR FAVORITE NUMBER IS ";
40 PRINT A
50 END
RUN
```

When the computer encounters an INPUT statement, it prints a question mark, and then waits for the user to type in an answer.

Type any number and then press RETURN .

String and numeric variables are not interchangeable. If you were to type B instead of 7, for example, an error message would appear on the screen.



Of course, you can rewrite the program to accept a word by using a string variable. Type:

```
NEW
05 GRAPHICS 0
10 DIM A$(30)
20 PRINT "WHAT IS YOUR FAVORITE COLOR?"
30 INPUT A$
40 PRINT "YOUR FAVORITE COLOR IS ";
50 PRINT A$
60 END
RUN
```

In both cases, line 30 (INPUT A or INPUT A$) assigns whatever number or word you have typed to variable A or A$ in memory, and then prints this value on the screen in line 50.

Notice that in line 10 the DIM (Dimension) statement reserves memory and assigns it to the variable A$. We have asked for 30 characters to be assigned. If you were to run the program again, this time giving a different answer to the question, "What's your favorite color?" the computer would assign a new value to the variable stored in memory.

A string variable can also be used to represent a combination of letters and numbers. Type:

```
NEW
05 GRAPHICS 0
10 DIM P$(10)
20 PRINT "WHAT'S YOUR POSTAL ZIP CODE?"
30 INPUT P$
40 PRINT "YOUR POSTAL ZIP CODE IS";
50 PRINT P$
60 END
RUN
```

SUMMARY

There are two types of instructions: system commands and program commands. System commands, such as LIST, RUN and NEW, tell the computer to do something with or to a program. Program commands, such as PRINT and INPUT, are contained within the body of a program.

Numeric and string variables allow the computer to ask the user for some information. Numeric variables are used to represent numbers; string variables, to represent letters, groups of numbers, or a combination of letters and numbers.

UNIT V
PROGRAMMING IN BASIC (2)

ASSIGNING VALUES

Numeric and string variables can also be assigned specific values. In the following example, A is defined as 15, B as 7, and C as A*B or 15 x 7. Type:

```
NEW
05 GRAPHICS 0
10 A=15
20 B=7
30 LET C=A*B
40 PRINT C
50 END
RUN
```

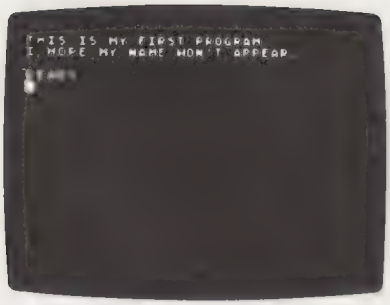
Now type:

```
NEW
05 GRAPHICS 0
10 DIM A$(19),B$(24)
20 A$="10,20,30,40 . . . . . 100"
30 B$="THIS IS COUNTING BY 10'S."
40 PRINT A$
50 PRINT B$
60 END
RUN
```

REM

REM statements are remarks or comments that do not show up on the screen when the program is run, but do appear in a listing of the program. Type:

```
NEW
05 GRAPHICS 0
10 REM (YOUR NAME)
20 REM (TODAY'S DATE)
30 PRINT "THIS IS MY FIRST PROGRAM."
40 PRINT "I HOPE MY NAME WON'T APPEAR."
50 END
RUN
```

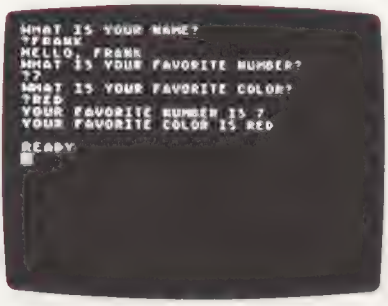


REM statements can be placed anywhere in a program. They are often used to identify the programmer and the program.

REM statements are especially useful when it comes to modifying or revising a program. (More about this in UNIT VI.)

In the next example, REM statements are used to remind you of the value of the input variables. Type:

```
NEW
 05 GRAPHICS 0
 10 DIM N$(30),B$(30)
 20 PRINT "WHAT IS YOUR NAME?"
 30 INPUT N$
 40 PRINT "HELLO, ";
 50 PRINT N$
 60 PRINT " WHAT IS YOUR FAVORITE NUMBER?"
 70 INPUT A
 80 PRINT "WHAT IS YOUR FAVORITE COLOR?"
 90 INPUT B$
100 REM N$ IS THE NAME
110 REM A IS THE NUMBER
120 REM B$ IS THE COLOR
130 PRINT "YOUR FAVORITE NUMBER IS ";
140 PRINT A
150 PRINT "YOUR FAVORITE COLOR IS ";
160 PRINT B$
170 END
RUN
```



GOTO

GOTO is a program command that is used to send the computer to a line other than the very next in the program. Type NEW and press RETURN . Then type:

```
05 GRAPHICS 0
10 PRINT "ADDITION"
20 PRINT "2 + 3 = "
30 INPUT C
40 IF C = 5 THEN GOTO 70
50 PRINT "WRONG. TRY AGAIN."
60 GOTO 20
70 PRINT "GOOD WORK!"
80 END
RUN
```

If the user answers the question incorrectly, the computer prints line 50: WRONG. TRY AGAIN. Line 60 tells the computer to go back to line 20: 2 + 3 = .

GOTO does just what it says — it tells the computer to go to a specific line. In the program you just typed, the GOTO statement creates a "loop" by telling the computer to go back to line 20. If you have not already done so, run the program.

A GOTO loop can run indefinitely. Type NEW and then press RETURN

Then type:

```
05 GRAPHICS 0
10 PRINT "HELP! MY COMPUTER HAS GONE BERSERK!"
20 GOTO 10
30 END
RUN
```

This is called an infinite loop. You can stop it by holding down the BREAK key. This will cause a "break" message to appear on the screen.

IF. . .THEN

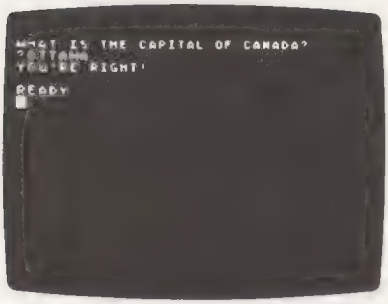
IF. . .THEN statements are among the most powerful relating to decision making. They can be interpreted thus: if a particular situation is true, then the computer will act in a specific way; if a particular situation is not true, then the computer will not act in this way.

Type NEW and press RETURN to clear the computer's memory. Then type:

```
02 GRAPHICS 0
05 DIM C$(6)
10 PRINT "WHAT IS THE CAPITAL OF CANADA?"
20 INPUT C$
30 IF C$ = "OTTAWA" THEN 60
40 PRINT "NO. TRY AGAIN."
50 GOTO 10
60 PRINT "YOU'RE RIGHT!"
70 END
```

In line 30, the computer compares your answer (C$) with OTTAWA. If C$ = OTTAWA, then the computer jumps to line 60 and prints YOU'RE RIGHT. If C$ does not = OTTAWA, then the computer goes to line 40 and prints NO. TRY AGAIN. Line 50 tells the computer to go to line 10, which prints the question, WHAT IS THE CAPITAL OF CANADA?

Now, run the program.



There is one flaw in this program. Perhaps you already spotted it. If the user does not type OTTAWA *exactly* as it appears in line 30, the computer will print NO. TRY AGAIN. In other words, you cannot type Ottawa or ottawa; the computer will only accept OTTAWA.

REVIEW

Let's review some of the new BASIC commands you learned in this unit.

Type NEW and press RETURN. Then type:

```
 05 GRAPHICS 0
 10 REM THIS IS A REVIEW
 15 DIM A$(3),N$(30)
 20 PRINT "WHOSE PROGRAM IS THIS?"
 30 INPUT N$
 40 PRINT "DOES IT WORK?"
 50 PRINT "ANSWER YES OR NO, "; N$
 60 INPUT A$
 70 IF A$="YES" THEN 90
 80 GOTO 110
 90 PRINT "GOOD WORK, "; N$
100 GOTO 120
110 PRINT "TRY AGAIN, "; N$
120 END
RUN
```

An INPUT statement is used when a programmer wants the user to enter data during the execution of a program. A question mark will appear on the screen to let the user know that it is his or her turn to type in some information.

The N$ that appears in lines 30, 50, 90, and 110 is called a string variable. It allows the user to enter letters in groups or "strings" that the computer will read, remember, and print as instructed.

The semicolon (;) that appears in front of the string variable in lines 50, 90, and 110 tells the computer to print the value of N$ on the same line.

The variable statement IF A$="YES" THEN 90 appears in line 70. This means, if the user answered YES to the question in line 40, then the computer will print GOOD WORK. A GOTO statement transfers control in the program without any conditions at all. An IF. . .THEN statement transfers control only if certain conditions are true.

Here's another sample program. Remember to clear the computer's memory first by typing NEW and pressing RETURN.

```
 02 GRAPHICS 0
 05 REM THIS IS A REVIEW
 10 DIM N$(30),Y$(1)
 15 PRINT "MULTIPLICATION CALCULATIONS"
 20 PRINT "WHAT IS YOUR NAME?"
 30 INPUT N$
 40 PRINT "TYPE IN TWO NUMBERS, ";
 45 PRINT N$
 50 PRINT "SEPARATE THEM WITH A COMMA."
 60 INPUT A,B
 70 PRINT "WHAT IS"; A; "*"; B
 80 INPUT C
 90 IF C=A*B THEN 120
100 PRINT "NOT QUITE. THE ANSWER IS "; A*B
110 GOTO 130
120 PRINT "EXCELLENT!"
130 PRINT "DO YOU WANT TO TRY ANOTHER? Y/N"
140 INPUT Y$
150 IF Y$="Y" THEN 40
160 PRINT "THAT'S ALL, ";
165 PRINT N$
170 END
RUN
```

SUMMARY

REM allows you to enter statements that will not be seen when the program is run, but will appear on the screen when you list the program.

GOTO statements tell the computer to go to a line other than the very next in the program. IF. . .THEN statements tell the computer to act in a specific way only if certain conditions are true.

UNIT VI
HOW TO MODIFY A PROGRAM

In this unit, we'll look at two program commands that can be used to customize or modify programs to suit your own purposes: PRINT and READ. . .DATA. You've already learned about print statements. READ. . .DATA is a command that tells the computer to read numeric or alphabetic data from DATA statements. A DATA statement contains information that will be used in a program (e.g., the correct answers to multiple choice or addition questions). A DATA statement is always paired with a READ statement.

Each time the computer reads and stores a new piece of data from the DATA statement, it then goes back and reads the next piece of data, and then the next after that. When the computer has read enough information, it continues with the program.

Here's a program for you to type. Be sure to empty the computer's memory first; type NEW and press RETURN . Then, clear the screen by pressing SHIFT and CLEAR simultaneously.

```
 02 GRAPHICS 0
 05 REM THIS IS A MULTIPLICATION DRILL.
 10 DIM F$(30),A$(30),E$(30)
 15 PRINT "THIS IS A MULTIPLICATION DRILL."
 20 PRINT "IT WILL TEST THE 5X TABLE."
 30 PRINT "TYPE IN THE ANSWER."
 35 PRINT
 40 REM F$ REPRESENTS THE QUESTION
 45 REM E$ REPRESENTS THE ANSWER
 50 REM A$ IS YOUR ANSWER
 60 FOR I=1 TO 10
 70 READ F$,E$
 80 PRINT F$;" = ";:INPUT A$
 90 IF A$=E$ THEN 110
100 PRINT "NO, THE ANSWER IS ";E$
110 NEXT I
120 PRINT "THAT'S ALL."
300 DATA 5*1,5,5*2,10,5*3,15,5*4,205*5,25
310 DATA 5*6,30,5*7,35,5*8,40,5*9,45,5*10,50
400 END
RUN
```

LISTING THE PROGRAM

Let's go back and take a closer look at the program you just typed. Type:

```
LIST 5,35
```

Notice the REM statement in line 05; it did not appear on the screen when you ran the program. It only shows up in a listing. Lines 15 to 35 are all PRINT statements: whatever appears within quotation marks will appear on the screen. Look closely at line 35. Nothing appears beside the word PRINT. If you were to run the program again, you would see that this gives a blank line. This is yet another use of the BASIC command PRINT: to leave space on the screen.

Now type:

LIST 40,100

Notice the two REM statements. These tell you that F$ and E$ are the string variables that represent the questions and correct answers, respectively. In this program the I that appears in line 40 is the counter; that is, it keeps track of the number of questions that have appeared on the screen.

Clear the screen by pressing CTRL and CLEAR and press RETURN . Then type:

LIST 80,310

Line 80 tells the computer to print the first question in the DATA statement (F$), and tells it to wait for the user to input an answer (A$). Line 90 tells the computer that if A$ = E$, it should then go to line 110, which tells the computer to present the next question (I), until all 10 questions have appeared on the screen. If A$ does not equal E$, then the computer goes to line 100, which tells it to print NO and to give the correct answer.

The program continues in this way until all 10 multiplication questions have appeared on the screen. Notice that the program uses the DATA statements (lines 300 and 310) in the order in which they appear. That is, the first question to appear on the screen is always 5*1, followed by 5*2, 5*3, etc. This limits the program's educational value, since users can easily memorize the order of the questions and their answers. There are programming techniques to remedy this, but they do not come within the scope of this manual.

MODIFYING A PROGRAM

The multiplication program discussed above can be adapted to a variety of different purposes. Say, for example, you want to turn it into a geography drill that tests the names of countries and their capitals. What needs to be changed?

In line 05 you would have to change "multiplication" to "geography."

Line 15 would be: PRINT "THIS IS A TEST OF COUNTRIES"; Line 20: PRINT "AND THEIR CAPITALS,"; and line 30: PRINT "NAME THE CAPITALS."

Similarly, the DATA statements would have to be changed to reflect the names of the countries and their capitals.

In order to make all these changes, you need only to type the line number, followed by the new information. Type:

```
05 REM THIS IS A GEOGRAPHY DRILL.
10 DIM F$(100),A$(100),E$(100)
15 PRINT "THIS IS A TEST OF COUNTRIES"
20 PRINT "AND THEIR CAPITALS."
30 PRINT "NAME THE CAPITALS."
300 DATA NEW ZEALAND, WELLINGTON, POLAND,
    WARSAW, UNITED ARAB EMIRATES, ABU DHABI
310 DATA THAILAND, BANGKOK, EQUADOR, QUITO,
    PORTUGAL, LISBON
320 DATA PHILIPPINES, MANILA, BOLIVIA, LA PAZ,
    ZIMBABWE, SALISBURY, ZAMBIA, LUSAKA
```

Since there are 10 multiplication questions in the original program (i.e., 10 pairs of questions and answers), there can only be 10 geography questions (i.e., 10 pairs of countries and capitals); otherwise, the computer wouldn't be able to match the data evenly.

The computer automatically substitutes the new information for the old information.

Note the change from 30 characters to 100 characters in the DIM statement. This is because each of the place names occupies more memory space than the previous sets of figures in the multiplication problems.

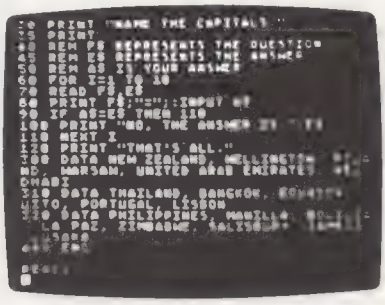
To correct the references in the REM statements, type:

```
45 REM F$ REPRESENTS THE COUNTRIES
75 REM E$ REPRESENTS THE CAPITALS
```

To check that you have made these changes correctly, type:

```
LIST 45,75
```

Now, run the program.



SUMMARY

PRINT and READ. . .DATA statements can be used to modify programs. The command PRINT followed by no information will cause a blank line to appear in a program.

Computer programs can be modified to suit a number of different purposes. Remember, though, the number of pieces of DATA in the revised program must equal the number that appeared in the original program. Otherwise, the DATA statements will not match up evenly.

To make corrections in an existing program, simply retype the line. To delete a line from an existing program, type the line number, followed by no information.

TROUBLESHOOTING

To clear the screen, (1) press CTRL and CLEAR , or (2) press SHIFT and CLEAR , or (3) type GRAPHICS 0 and press RETURN .

To empty the computer's memory, (1) type NEW and press RETURN , or (2) turn the computer off and on again.

To correct a typing error, use the cursor control keys, the DELETE BACK S key and the CTRL and INSERT or DELETE keys.

To correct an error in a program, simply retype the line.

To delete a line from a program, simply type the line number with no information beside it.

To insert a line in a program, simply type the line, beginning with a number to indicate its placement. The computer will automatically position the line in the correct place in the sequence.

To leave a blank line in a program, type PRINT with no information beside it.

To stop a program in the middle of its run, press BREAK .

EXPLANATION OF ERROR MESSAGES

ERROR 8 — INPUT STATEMENT ERROR. If this message appears on the screen, you have typed an alphabetic letter to the input request when a numeric variable was used. The INPUT statement would only accept a number.

ERROR 9 — ARRAY OR STRING DIM ERROR. When this message appears on the screen, (1) you DIMensioned a string bigger than the computer's memory would allow, or (2) you tried to DIMension an array that had previously been DIMensioned, or (3) you attempted to reference a string or array that had not been DIMensioned.

ERROR 12 — LINE NOT FOUND. You tried to have the computer GOTO a line number that didn't exist. Check your program listing and compare it to what you typed in.

ERROR 13 — NO MATCHING FOR STATEMENT. If this message appears on the screen, you have probably left out a FOR statement or a NEXT statement, in the program. LIST the program and check for omissions.

ERROR 17 — GARBAGE ERROR. This message means that you have failed to correct a line that had a syntax error in it. LIST the line and retype it.

ERROR 138 or ERROR 143 — These error messages mean that something has gone wrong during a LOAD. Rewind the cassette and try to load the program again.

GLOSSARY

address: Every letter, symbol, and piece of data is placed somewhere in the computer's memory. This memory location has an address, which is a number.

BASIC: Beginner's All-purpose Symbolic Instruction Code. This is a computer language commonly used on microcomputers.

branching: When a program is being executed, certain instructions will cause a transfer of control to another line.

break: When a program is running, you can stop it. The message that appears on the screen is STOPPED AT (number). This means that you stopped the program at a particular line number.

byte: A group of eight bits. A byte is generally considered to represent one character.

control: Either you or the computer is in control. This means that either it is your turn to enter data, or the computer's turn to execute a program.

cursor: This is the white box that indicates that the computer is awaiting your next instruction.

debugging: Whenever there are errors in a program, you must find and correct them. This is known as debugging.

edit: To prepare data for subsequent processing, e.g., by the deletion of errors.

END: All programs in BASIC must have an END statement.

execute: To perform the operations specified by an instruction. The program will execute the commands within it once it has been given the RUN command.

GOTO: A branch instruction in the BASIC language.

hardware: The actual physical components of a computer system.

IF...THEN: A program statement that passes control to a different line, if a particular condition has been met.

immediate execution: When the computer immediately performs the commands we give it. When the computer is being used as a calculator or to create graphics without a program, it is in immediate execution mode.

line: Every line must have a number identifying it in a BASIC program. This group of digits is numbered sequentially, but not necessarily consecutively.

loop: A group of instructions that is executed more than once.

machine code: This is the language actually used by computers to carry out instructions.

memory: A storage area for data within the computer.

microcomputer: A complete, small computer system.

numeric variable: A variable that represents a number.

peripheral: Any device connected to a computer that is to some degree controlled by the computer.

program: A sequence of instructions that results in the execution of an algorithm.

?: A short form for PRINT.

RAM: Random Access Memory is usable or working memory. Anything stored here is lost when the machine is turned off.

REM: This stands for REMark. In a program, a REM statement gives information, but it will not be printed on the screen.

RETURN: A RETURN key on the computer is used to enter information. On some computers, the same function may be performed by an ENTER key.

ROM: Read Only Memory keeps permanent information within itself.

scroll: This refers to when the contents of the screen move up or down by one or more lines.

software: The instructions that tell hardware what to do with data, i.e., programs.

statement: A command or set of commands in one line of a program.

string variable: A variable used to represent strings of characters, such as a word.

syntax error: An error message that often appears in BASIC programs.

system command: A command that will cause the computer to do something immediately, e.g., SAVE, LOAD, etc.

type mismatch error: An error message that occurs when the expected response is not given in a program.


WNET